AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (currently amended) A memory system for a computer, the memory system comprising a single memory page including a kernel stack, [[and]] a register stack engine (RSE) stack, and a data structure comprising system information about a user process, ~~wherein~~ the kernel stack [[is]] being separate and distinct from user program stacks in the memory system, wherein the kernel stack and the RSE stack are separated by said data structure, and wherein the kernel stack and the RSE stack are configured to grow apart in opposite directions away from said data structure.

2. (canceled)

3. (canceled)

4. (canceled)

5. (currently amended) The memory system of ~~claim 2~~ claim 1, further comprising:

a first red zone in a second memory page bordering a first memory region of the single memory page which is allocated to the kernel stack.

6. (original) The memory system of claim 5, further comprising:

a second red zone in a third memory page bordering a second memory region of the single memory page which is allocated to the RSE stack.

7. (currently amended) The memory system of ~~claim 3~~ claim 1, wherein a number of translation lookaside buffer (TLB) misses when a process "enters" a kernel of an operating system of the computer is no more than one TLB miss.

8. (currently amended) The memory system of ~~claim 2~~ claim 1, wherein the memory system further comprises a stack overflow handler that is configured to allocate more memory to one of the stacks if it overflows.

9. (original) The memory system of claim 1, wherein the memory system is used in cooperation with an operating system for the computer, and wherein the operating system comprises a flavor of UNIX.

10. (previously presented) The memory system of claim 1, wherein the memory system is used in cooperation with at least one microprocessor with an Itanium Processor Family processor architecture.

11. (currently amended) A computer system comprising:

a microprocessor including a register stack and a register stack engine (RSE);

an operating system including a kernel; and

a memory system configured to have a single memory page that includes [[both]] a kernel stack, [[and]] an RSE stack, and a data structure comprising system information about a user process, ~~wherein~~ the

kernel stack [[is]] being separate and distinct from user program stacks in the memory system,

wherein the kernel stack and the RSE stack are separated by said data structure, and wherein the kernel stack and the RSE stack are configured to grow apart in opposite directions away from said data structure.

12. (canceled)

13. (canceled)

14. (canceled)

15. (currently amended) The computer system of ~~claim 12~~ claim 11, further comprising:

a first red zone in a second memory page bordering a first memory region of the single memory page which is allocated to the kernel stack.

16. (original) The computer system of claim 15, further comprising:

a second red zone in a third memory page bordering a second memory region of the single memory page which is allocated to the RSE stack.

17. (currently amended) The computer system of ~~claim 13~~ claim 11, wherein a number of translation lookaside buffer (TLB) misses when a process "enters" a kernel of the operating system is no more than one TLB miss.

18. (currently amended) The computer system of ~~claim 13~~ claim 11, wherein the memory system further comprises a stack overflow handler that is configured to allocate more memory to one of the stacks if it overflows.

19. (previously presented) The computer system of claim 11, wherein the microprocessor is configured with an Itanium Processor Family processor architecture.

20. (original) The computer system of claim 11, wherein the operating system comprises a flavor of UNIX.

21. (currently amended) A method of a process entering a kernel of an operating system configured for an Itanium Processor Family processor architecture, the method comprising:

accessing a kernel stack within a memory page;

accessing an RSE stack within the same memory page; and

accessing a [[uarea]] data structure within the same memory page,

wherein the kernel stack is separate and distinct from user program stacks,

wherein said data structure comprises system information about a user process, and

wherein the kernel stack and the RSE stack are configured to grow apart in opposite directions away from said data structure.